

REMARKS

Claims 1-17 and 20-33 were pending in the application, and the Examiner rejected all claims. Applicant respectfully traverses the Examiner's rejection.

On pages 2-6, the Examiner rejected claims 1-17, 20-23 and 32-33 under 35 U.S.C. §103(a) as being unpatentable over Loatman et al. (U.S. Patent No. 4,914,590) in view of a newly cited reference, Hicken et al. (U.S. Patent No. 6,473,896).

As acknowledged by the Examiner, Loatman et al. does not appear to teach or suggest that either rules which were unsuccessfully applied to the input text, or rules which were not applied at all to the input text in obtaining the displayed parse, are displayed to the user. As described in the background portion of the present application, this requires the user, when debugging the system, to remember all rules which could apply to the input text, and to choose one of those rules from the user's memory, to correct an incorrect parse. This does not appear to be addressed anywhere by Loatman et al.

Instead, the passages of Loatman et al. cited by the Examiner appear to show nothing more than displaying the rules which were actually and successfully applied to the input text to obtain the displayed parse. If the user is not satisfied with the parse, or if the parse is actually incorrect, the user is still faced with the same problem discussed with respect to prior art systems - that is, the user must rely on his or her memory to recall the rules which could be applied in a given situation, and to modify the parse by entering one of those alternate rules.

This deficiency is not remedied by Hicken et al. Hicken et al. has nothing to do with applying grammar rules to parse an input text to obtain a parse tree. Instead, Hicken et al. discusses a system for determining the accuracy of a language by determining whether the language violates any rules. Therefore, the rules discussed throughout Hicken et al. have nothing to do

with rules that are used to generate a parse tree, but are only rules that are examined, given the already generated parse tree, to determine whether the parse tree violates any of those rules. See table 1, columns 5-6. The rules represent patterns that are not allowed to appear in the source code. See column 8, lines 58-61.

This has nothing to do with determining whether the parse is correct given the input. Instead, the Hicken et al. system assumes that the parse is correct and simply checks the parse to see whether it violates any rules, in order to determine whether the underlying source code input is correct. Of course, this is a fundamentally different type of system than the present invention.

The specific figures and text cited by the Examiner discuss how to create a new constraint (or rule) to be added to the list of constraints in database 17. This has nothing whatever to do with creating a parse tree. Instead, it discusses how to create a representation of a pattern that is not allowed in the original input.

More specifically, a given node type in Hicken et al. can violate a number of different constraints (or rules) in database 17. Therefore, when creating a new constraint to be added to database 17, the cursor is placed over a node type and if that node type is selected, a display generates all the attributes of that node type. This is what is being illustrated in FIG. 8. In fact, the figures cited by the Examiner have nothing to do with generating a parse tree, and they have nothing to do with displaying rules that were applied, or not applied.

FIG. 9 illustrates that code that is executed when the given node type is identified can also be displayed. Similarly, a given node type can violate several different rules and all of the rules which are required by (or violated by) the node type can be displayed. See column 11, lines 47-51. Of course, this collective teaching simply shows that the attributes and rules

that a particular node type violates can be displayed. Again, this has nothing to do with grammar rules used to generate a parse tree. Nor does it have anything to do with displaying grammar rules that were applied or those that were not applied, when generating the parse tree. It simply does not relate to displaying rules of any kind that were, or were not, applied. It is a fundamentally different type of system.

By contrast, claim 1 specifically includes "displaying a parse tree for a textual input, the parse tree being generated based on rules and comprising at least one connecting point having two children... and displaying a plurality of display items proximate to said selected connecting point, the display items including alternate rules, other than the rules used in generating a constituent at the selected connecting point." Similarly, independent claim 10 states "generating a parse tree, generated by applying grammar rules, for an input text comprising at least one connecting point having two children... and displaying a first plurality of menu items proximate to said selected connecting point, the menu items including an alternate grammar rules display item which, when activated, displays alternate grammar rules comprising grammar rules not successfully applied in generating the parse tree." Neither of the references cited by the Examiner, either alone or in combination, either teach or suggest these features of the present invention. Therefore, Applicant submits that independent claims 1 and 10 are allowable over the references cited by the Examiner.

At the bottom of page 6 of the Office Action, the Examiner made the bald statement that claims 24-31 are the same in scope and content as claims 1-23 and are therefore rejected under the same rationale. This is simply inaccurate. The Examiner made no detailed rejection of claims 24-31. Therefore, Applicant is unable to respond in detail to any such rejection. Applicant would simply point out that the limitations in independent claims

24 and 29 are not found in the previous claims and therefore Applicant will attempt to clarify the patentability of those claims, even without a detailed rejection from the Examiner.

Independent claim 24 specifically states "generating a plurality of different parse trees for a textual input; displaying one of the plurality of parse trees... receiving control input selecting a new parse... and... displaying another of the plurality of parse trees." Thus, with a plurality of parse trees generated for a single given input text, the system allows the user to select, for display, different ones of the parse trees by simply providing a control input. This is not found in claims 1-23, as stated by the Examiner, and it is simply neither taught nor suggested by any of the references cited by the Examiner.

It does not appear that Loatman et al. addresses this in anyway. More specifically, it does not appear that Loatman et al. teaches or suggests that multiple parses are generated for a given input text. Similarly, there is no indication in Hicken et al. that a parse tree is ever even displayed, much less multiple parse trees for the same input. Therefore, neither of the references cited by the Examiner, either alone or in combination, teach or suggest the invention set out in independent claim 24. Thus, Applicant respectfully submits that independent claim 24 is allowable over the references cited by the Examiner.

Finally, independent claim 29 states "receiving user control input selecting a new textual input to be parsed... and displaying said new parse of said new textual input as a new parse tree". Thus, the system allows a user to rapidly switch among the various training sentences in a corpus, and for having each of those sentences parsed simply by clicking a control input. This is not found in claims 1-23, as suggested by the Examiner.

Similarly, it is neither taught nor suggested by either of the references cited by the Examiner. Applicant has been unable to find any reference to this type of system in Loatman et al.,

and the Examiner has cited none. Nor is this taught or discussed, in anyway, by Hicken et al. Hicken et al. simply states that the input source code is parsed using a conventional parser. See column 4, line 66-column 5, line 5. There is no mention of receiving a user control input selecting a new textual input to be parsed and displaying a new parse tree for the new textual input.

Thus, Applicant submits that independent claim 29 is allowable over the references cited by the Examiner.

In sum, Applicant submits that independent claims 1, 10, 24 and 29 are allowable over the references cited by the Examiner. Applicant further submits that dependent claims 2-9, 11-17, and 20-33, which depend either directly or ultimately from the independent claims are allowable as well. Reconsideration and allowance of claims 1-17 and 20-33 are respectfully requested.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By:   
\_\_\_\_\_  
Joseph R. Kelly, Reg. No. 34,847  
Suite 1600 - International Centre  
900 Second Avenue South  
Minneapolis, Minnesota 55402-3319  
Phone: (612) 334-3222 Fax: (612) 339-3312

JRK:slg